Electronic Notes in Theoretical Computer Science 20 (1999)
URL: http://www.elsevier.nl/locate/entcs/volume20.html 15 pages

Towards a Strand Semantics for Authentication Logic

Paul Syverson [1]

Center for High Assurance Computer Systems
Naval Research Laboratory
Washington, DC , USA

Abstract

The logic BAN was developed in the late eighties to reason about authenticated key establishment protocols. It uncovered many flaws and properties of protocols, thus generating lots of attention in protocol analysis. BAN itself was also subject of much attention, and work was done examining its properties and limitations, developing extensions and alternatives, and giving it a semantics.

More recently, the strand space approach was developed. This approach gave a graph theoretic characterization of the causally possible interactions between local histories (strands) along with a term algebra to express sent and received messages. This model was designed and has been used by its authors for direct application to authentication protocol analysis. However, it has also quickly attracted the attention of many other researchers in the field as useful in connection to related work, such as model checking approaches.

Here we discuss the idea of using strand spaces as the model of computation underlying a semantics for BAN-style expressions. This will help to integrate some of the approaches to security protocol analysis and to hopefully provide BAN logics with a clearer, more useful underlying model than they have had to date.

1 Early Approaches to Knowledge

Automated approaches using model checkers, theorem provers and the like have increasingly been at the heart of formal analysis of security protocols for the last several years. However, for much of the nineties the most well known and successful approach to this problem was by hand analysis using specialized logics. A belief logic, BAN [2], was widely used to reveal a number of flaws and

[1] This work supported by NSA and ONR. Thanks to Cathy Meadows for many helpful comments and discussions.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
1999		**00-00-1999 to 00-00-1999**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Towards a Strand Semantics for Authentication Logic	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Research Laboratory,Center for High Assurance Computer Systems,4555 Overlook Avenue, SW,Washington,DC,20375**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**15**	

hidden assumptions in protocols. It also gave rise to a number of extensions, variations and related alternatives, which also had many successes. We will return to BAN below, but we first begin at the beginning. Hand logics themselves began to be published at about the same time as other formal methods of protocol analysis, in the late eighties. But, the first epistemic treatment of protocols may be found about five years earlier in the work of Merritt and various coauthors [10,4,11]. (As implied "hand logics" were originally devised for hand analysis; however, most of them have been automated in one form or another, often with great success.) Merritt's approach was basically algebraic rather than logical. However, this algebraic approach was used to characterize the state of knowledge of various protocol participants. We will see that this approach can be closely related to the semantics of an epistemic logic.

1.1 Algebraic Knowledge Semantics

The semantics used to underly much of epistemic logic is based on a model theoretic treatment of possible worlds. The idea of possible worlds is that there are different ways the world might be (or epistemically, might be conceived to be). For each principal we can partition up all the possible worlds into those that are indistinguishable to him. Since this is a partition, it yields an indistinguishability relation that is an equivalence. For example, if Bob does not know whether Alice is in her office, then worlds in which Alice is in her office and worlds where she is not in her office are indistinguishable to him (excluding other distinguishing information). Thus, possible worlds can be used to underly a logic of knowledge. This has been studied as far back as [8]. The characterization of knowledge by equivalence relations is in fact just one of the types of knowledge set out in [8] and later. Other relations are possible; thus indistinguishability is sometimes more generally called 'accessibility'—e.g., the relation might not be symmetric. What has all this to do with cryptographic protocols?

Suppose Alice and Bob are executing a coin-flip protocol. Alice sends to Bob two messages in random order, one is the encryption of a bit representing a heads and the other that of a bit representing a tails. They are both encrypted with the same key, which is known only to Alice. In the common notation we have, $\{Heads\}_K$ and $\{Tails\}_K$ where Bob does not know K. Bob does not know whether the first or second message is the encryption of *Heads*. (Actually, he does not know that either is the encryption of *Heads* at all. But, we ignore this for the moment.) So, there are (at least) two possible worlds indistinguishable by Bob: one where he has been sent the encryption of *Heads* followed by the encryption of *Tails*, and the other where that order is reversed.

Merritt examined such protocols using free algebras of messages with encryption and decryption operators. Such a free algebra represents the basic structure of the cryptosystem. The specific encryption and decryption algorithms used and the domain of messages is called the crypto-algebra. If we

assume that there are no flaws in the crypto algorithms themselves, we can basically assume that the free algebra and the crypto-algebra are isomorphic. In the coin-flip example, Bob does not know about all the messages. Thus, there are different homomorphic mappings from the free algebra to the cryptoalgebra that are indistinguishable by Bob. These mappings are effectively the indistinguishable possible worlds for Bob in this state.

This algebraic approach was extended by Toussaint [23] to examine evolving knowledge in protocol executions. The connection between such algebraic approaches and epistemic logic was made explicit by Bieber [3] when he used the constructs of [11] to underly the semantics of his logic CKT5. There have been other algebraic approaches to authentication protocol representation and analysis, for example, using process algebras such as CSP and the spi calculus. We will not discuss these in this paper.

2 The BAN Family

We now turn to a particular family of logics, stemming from the BAN logic of Burrows, Abadi, and Needham. BAN was created for examining authenticated key distribution protocols. These are typically protocols that allow two parties to establish a key for a secure communication session. Since the parties do not usually have a pre-existing shared secret, these protocols rely on a trusted server (either online or offline) to facilitate the distribution and often to generate the session key. Typical goals of such protocols are thus that the parties share the key, that no-one else does, and that they know with whom they are sharing the key.

The contribution of BAN was to set out a logic in simple terms (notably belief, jurisdiction, freshness, and the goodness of a key for two named principals) that revealed hidden assumptions and flaws in protocols through quite simple hand analysis. Another contribution that BAN made was to reason about time, but only in the roughest terms. Specifically, they distinguished only between messages that were fresh, i.e., sent during the current epoch, and those that were not. This also proved to be a very useful balance of simplicity and expressiveness.

Here is an example of a BAN "message-meaning" rule.

$$\frac{P \models P \overset{K}{\longleftrightarrow} Q,\ P \triangleleft \{X\}_K}{P \models Q \mid\sim X}$$

This basically says that if P believes K is a good key for P to talk with Q and P receives X encrypted with K, then P believes that Q once said X. The rule assumes that P can recognize messages he produced himself.

Rather than set out all the rules of BAN, we will go through the concepts that were introduced in BAN and sometimes modified by others. We will also generally use the notation of AT [1] and SVO [17], which is closer to ordinary English.

Freshness

A message is fresh if it has not been part of a message sent prior to the current epoch. It is sufficient but not necessary for freshness that a message be unseen prior to the current epoch. A principal might generate a message earlier and not send it until the epoch begins.

Freshness is central to the notion of authentication. Just because someone once said that a key was good, does not mean that they would say so now. If a message is bound, e.g., cryptographically, to a fresh message, then it must itself be fresh. Freshness is typically insured by means of nonces (random numbers generated to be recognized later by those who generate them) or timestamps from a trusted source.

One limitation of BAN is that the only way to promote to the present epoch from P *said* M (given that *fresh*(M)) is to say that P *believes* M. BAN has no expression P *says* M. Amongst other things, this either (1) limits the promotion of once-said to recently-said (believed) messages to formulae rather than messages in general, or (2) gives rise to a somewhat counterintuitive notion of belief. (Briefly, a formula is essentially a message that expresses a proposition. We assume a language in which all formulae are messages, but not necessarily vice versa.) The *says* notation was introduced in [1].

Saying and Receiving

As the message meaning rule illustrates, one says not only the messages one sends, but also certain messages implicit in what one sends. Similarly for receiving, e.g., one receives the concatenates of a concatenated message. BAN generally does not express those messages simply possessed by a principal, as opposed to sent and received. For example, in contrast to the message meaning rule, assuming that P *received* $\{X\}_K$, if P *sees* K (whether or not P *believes* $P \overset{K}{\leftrightarrow} Q$), then P *sees* X, whether or not P *believes* (Q *said* X). This expressiveness was added in GNY [7], and to some extent in [1].

More importantly, BAN cannot distinguish between those message that are understood by a recipient, e.g., upon decryption, and those that are not. Which is not to say that, BAN did not address this question. It was simply explicitly limited to describing receipt of messages that could be understood. Notation and rules to represent and reason with recognizability were added in [7], and a systematic semantic treatment of comprehension of messages was introduced in [17].

The semantics presented in this paper will distinguish between whole messages that P *received* and messages that may be contained in these that P *got* . We will see below, in section 4.3.1, that these have a more extensional meaning than the recognizable messages of GNY or the comprehended messages of SVO. We will leave for future work discussion of those messages that P simply possesses.

Jurisdiction

One needs a way to promote a claim by a key server that K is good for

P and Q to speak ($P \overset{K}{\leftrightarrow} Q$) to the truth of this claim. This is the notion of jurisdiction. In BAN, one could express that if P *believes* Q *controls* φ and P *believes* Q *said* φ, then P *believes* Q *believes* φ. In AT, with its *says* construct, jurisdiction can be boiled down to its essentials: if Q *controls* φ and Q *says* φ, then φ. Belief is not necessary to express jurisdiction. In general, AT separated out the belief axioms from the other axioms, allowing a normal modal logic of belief and a model-theoretic possible world semantics for it.

Keys

BAN is expressive enough to reason about both public-key and secret-key authentication protocols. As noted, ability to directly express possession of keys, and reason accordingly was added in GNY and AT. In VO [12], the ability to reason about Diffie-Hellman key agreement was added to a version of GNY. Also added was the differentiation of public key use for signature and encryption. A semantics for all of these was given in [17]. For the remainder of this paper, we will limit ourselves to secret-key expressions for simplicity. We will also not talk about other cryptographic constructs, such as hashes. These are all left for future work.

This concludes our nutshell exposition of the main concepts formalized in BAN-style languages. It does not nearly address all of the issues that were engendered by BAN, nor all of the authors that discussed them. However, it does cover all of the main concepts formalized in BAN. So, it is adequate for purposes of a first attempt to sketch a strand semantics for a BAN-style language. We now turn to the presentation of the necessary background on strand spaces.

3 Strand Spaces

In this section, we sketch out some of the basic elements of strand spaces. We also discuss some small extensions to make the model richer and to allow it to serve as a semantics for a richer logic. We present here only as much of the model as is needed to understand its use as a semantics in the next section. Further details can be found in [19–22].

A strand is basically a local history of sent and received messages in a protocol run. A strand space is a collection of strands. A bundle is a graph that reflects a causally meaningful way that a set of strands might be connected.

The messages sent between principals are taken from an algebra A of terms. We will say more about the algebra presently. Terms can be signed, e.g., $+t$ or $-t$, to indicate sending and receiving of messages respectively. Let Σ be a set of strands and $(\pm\mathsf{A})^*$ be the set of all finite sequences of signed terms. The following definitions are taken from [22].

Definition 3.1 A *strand space* over A is a set Σ together with a trace mapping $tr : \Sigma \to (\pm\mathsf{A})^*$.

Definition 3.2 *Fix a strand space* Σ

(i) *A* node *is a pair* $\langle s, i\rangle$*, with* $s \in \Sigma$ *and* i *an integer satisying* $1 \leq i \leq$ *length*$(tr(s))$*. The set of nodes is denoted by* $\mathcal{N}$*. We will say the node* $\langle s, i\rangle$ *belongs to the strand* s*. Clearly, every node belongs to a unique strand.*

(ii) *If* $n = \langle s, i\rangle \in \mathcal{N}$ *then* $index(n) = i$ *and* $strand(n) = s$*. Define* $term(n)$ *to be* $(tr(s))_i$*, i.e. the* i*th signed term in the trace of* s*. Similarly,* $uns\_term(n)$ *is* $((tr(s))_i)_2$*, i.e. the unsigned part of the* i*th signed term in the trace of* s*.*

(iii) *There is an edge* $n_1 \rightarrow n_2$ *if and only if* $term(n_1) = +a$ *and* $term(n_2) = -a$ *for some* $a \in \mathsf{A}$*. Intuitively, the edge means that node* n_1 *sends the message* a*, which is received by* n_2*, recording a potential causal link between those strands.*

(iv) *When* $n_1 = \langle s, i\rangle$ *and* $n_2 = \langle s, i+1\rangle$ *are members of* $\mathcal{N}$*, there is an edge* $n_1 \Rightarrow n_2$*. Intuitively, the edge expresses that* n_1 *is an immediate causal predecessor of* n_2 *on the strand* s*. We write* $n' \Rightarrow^+ n$ *to mean that* n' *precedes* n *(not necessarily immediately) on the same strand.*

$\mathcal{N}$ together with both sets of edges $n_1 \rightarrow n_2$ and $n_1 \Rightarrow n_2$ is a directed graph $\langle \mathcal{N}, (\rightarrow \cup \Rightarrow)\rangle$.

Definition 3.3 Suppose $\rightarrow_{\mathcal{C}} \subseteq \rightarrow$; suppose $\Rightarrow_{\mathcal{C}} \subseteq \Rightarrow$; and suppose $\mathcal{C} = \langle \mathcal{N}_{\mathcal{C}}, (\rightarrow_{\mathcal{C}} \cup \Rightarrow_{\mathcal{C}})\rangle$ is a subgraph of $\langle \mathcal{N}, (\rightarrow \cup \Rightarrow)\rangle$. $\mathcal{C}$ is a bundle if:

(i) $\mathcal{C}$ is finite.

(ii) If $n_2 \in \mathcal{N}_{\mathcal{C}}$ and term(n_2) is negative, then there is a unique n_1 such that $n_1 \rightarrow_{\mathcal{C}} n_2$.

(iii) If $n_2 \in \mathcal{N}_{\mathcal{C}}$ and $n_1 \Rightarrow n_2$ then $n_1 \Rightarrow_{\mathcal{C}} n_2$.

(iv) $\mathcal{C}$ is acyclic.

Definition 3.4 If $\mathcal{S}$ is a set of edges, i.e. $\mathcal{S} \subseteq (\rightarrow \cup \Rightarrow)$, then $\prec_{\mathcal{S}}$ is the transitive closure of $\mathcal{S}$, and $\preceq_{\mathcal{S}}$ is the reflexive, transitive closure of $\mathcal{S}$.

The relations $\prec_{\mathcal{S}}$ and $\preceq_{\mathcal{S}}$ are each subsets of $\mathcal{N}_{\mathcal{S}} \times \mathcal{N}_{\mathcal{S}}$, where $\mathcal{N}_{\mathcal{S}}$ is the set of nodes incident with any edge in $\mathcal{S}$.

These are all of the definitions that we need to set out a possible worlds model and semantics for sending, receiving, and knowledge. We will provide below more details about the term algebra that will allow us to express, e.g., that a principal who receives a ciphertext (encrypted message) and has the decryption key has also got the unencrypted message.

4 Possible worlds from Strand Spaces

We now describe the possible world semantics of epistemic logics for distributed computing in general and for security protocols in particular, e.g., [1,17].

4.1 Traditional System Model and Knowledge Semantics

Computation is performed by a finite set of principals, $P_1, \ldots, P_n$, who send messages to one another. In addition there is a principal P_e representing the environment. This allows modeling of any penetrator actions as well as reflecting messages in transit.

Each principal P_i has a local state s_i. A global state is thus an $(n+1)$-tuple of local states.

A run is a sequence of global states indexed by integral times. The first state of a given run r is assigned a time $t_r \leq 0$. The initial state of the current authentication is at $t = 0$. The global state at time t in run r determines a possible world (sometimes also called nodes or points). We assume that global states are unique wrt runs and times. Thus, they can be referred to by, e.g., '$\langle r, t\rangle$'. At any given global state, various things will be true, e.g., that principal Q has previously sent the message $\{X\}_K$. What a principal P then knows (believes) at a given point $\langle r, t\rangle$ is precisely that which is true at all possible worlds with the same local state $r_P(t)$ for P as $\langle r, t\rangle$. This is typically captured by means of an accessibility relation on global states $\rightsquigarrow_P$ for a principal P. As noted in section 1.1, when the relation is an equivalence, it is also called an indistinguishability relation $\sim_P$ for a principal P. This allows for a simple intuitive definition, without even having to describe in any way properties of local states, viz:

- $\langle r, t\rangle \sim_P \langle r', t'\rangle$ iff P is in the same local state at both points, i.e., $r_P(t) = r'_P(t')$.

(**Aside:** When the relation is not an equivalence, we typically need to say something about properties of the local state to describe the relation, although not necessarily much. For example, if we have some meaningful notion of substate, one possibility is

- $\langle r, t\rangle \rightsquigarrow_P \langle r', t'\rangle$ iff $r'_P(t')$ is a substate of $r_P(t)$

The accessibility relation we will set out below is an equivalence, and we will say no more about this.)

Given an indistinguishability relation, we can then go on to define principal P's knowledge in terms of the worlds that are P-indistinguishable.

- $\langle r, t\rangle \models P$ *knows* φ iff $\langle r', t'\rangle \models \varphi$ for all $\langle r', t'\rangle$ such that $\langle r, t\rangle \sim_P \langle r', t'\rangle$

(**Aside:** We have sketched out a semantics for knowledge, specifically **S5** knowledge, for a distributed system. The modality for most of the logics in the BAN family is in fact belief. The reasons for and significance of choosing one or the other have been discussed elsewhere, e.g. [1,14], and we will say no more about the matter here.)

The above system model and characterization of knowledge is essentially what is found in [1,17]. It is largely based on similar models and characterizations of knowledge in distributed computing. (Cf., e.g., [6].) We now turn

specifically to strand spaces as a basis for knowledge semantics.

4.2 Strand Semantics for Knowledge

In the conclusion of [19] it was suggested that, "[w]hat a protocol participant knows, in virtue of his experience in executing a protocol, is that he has performed the actions lying on some strand s. Thus, the real world must include some bundle $\mathcal{C}$ such that s is contained in $\mathcal{C}$. The beliefs that the participant may justifiably hold are those that are true in every bundle $\mathcal{C}$ containing s."

Thus, a possible world on this approach is simply a bundle. This is a reasonable approach for reasoning about some protocol features. However, we found it also worthwhile to include in the definition of possible worlds the nodes within bundles. We did this in order to capture temporal aspects of the above authentication logics, specifically freshness. (This will also facilitate the addition of richer temporal formulae to the logic, as in [15].)

Neither strand spaces nor bundles have a notion of global time. Thus we cannot have an indistinguishability relation that corresponds directly to the above. However, $\langle \mathcal{C}, s, i \rangle$ picks a unique point $\langle s, i \rangle$ in bundle $\mathcal{C}$ and partitions $\mathcal{N}_\mathcal{C}$ into $\{\langle t, j \rangle : \langle t, j \rangle \preceq_\mathcal{C} \langle s, i \rangle\}$ and $\{\langle t, j \rangle : \langle t, j \rangle \not\preceq_\mathcal{C} \langle s, i \rangle\}$. This partition allows us to define an accessibility relation on nodes in bundles based on local time.

Definition 4.1 (i) Given a strand s, let $\mathrm{princ}(s)$ refer to the principal whose strand s is.

(ii) Given a node $\langle s, i \rangle$ and a strand t in a bundle $\mathcal{C}$, let the *restriction of* t *to* $\langle s, i \rangle$ *in* $\mathcal{C}$ be $\mathrm{tr}(t) \restriction \langle s, i \rangle = \langle \mathrm{tr}(t)_1, \ldots, \mathrm{tr}(t)_j \rangle$, where $\langle t, j \rangle$ is the greatest node on t s.t. $\langle t, j \rangle \preceq_\mathcal{C} \langle s, i \rangle$.

With this notation in place we can now define an indistinguishability relation.

Assume bundles $\mathcal{C}, \mathcal{C}'$, and strands s, s', and indices i, i' such that $\langle s, i \rangle \in \mathcal{N}_\mathcal{C}$ and, $\langle s', i' \rangle \in \mathcal{N}_{\mathcal{C}'}$. A natural definition, analogous to the runs-and-times definition of the traditional literature would be to have $\langle \mathcal{C}, s, i \rangle \sim_P \langle \mathcal{C}', s', i' \rangle$ (i.e., $\langle \mathcal{C}, s, i \rangle$ is *P-indistinguishable* from $\langle \mathcal{C}', s', i' \rangle$) just in case P's history in $\mathcal{C}$ up to $\langle s, i \rangle$ matches P's history in $\mathcal{C}'$ up to $\langle s', i' \rangle$. This is exactly right. However, just as there is no global time in a bundle, there may also be multiple strands associated with one principal. The resulting definition is thus:

Definition 4.2 $\langle \mathcal{C}, s, i \rangle$ is *P-indistinguishable* from $\langle \mathcal{C}', s', i' \rangle$
(written as $\langle \mathcal{C}, s, i \rangle \sim_P \langle \mathcal{C}', s', i' \rangle$) iff

(i) for any t in $\mathcal{C}$ s.t. $\mathrm{princ}(t) = P$ there exists t' in $\mathcal{C}'$ s.t. $\mathrm{tr}(t) \restriction \langle s, i \rangle = \mathrm{tr}(t') \restriction (s', i')$ and $\mathrm{princ}(t') = P$, and

(ii) the number of strands satisfying clause i is the same in $\mathcal{C}$ and $\mathcal{C}'$.

4.3 Truth Conditions for BAN-Style Formulae

The purpose of this section, is to present truth conditions for basic formulae of a BAN-style language. The basic notions we cover are freshness, key goodness, said and received (got) messages, and jurisdiction.

Given our definition of $\sim_P$ above we can now present truth conditions for knowledge in this semantics. Let φ be some formula in our language. We will define $\models$ inductively; however the presentation is organized pedagogically rather than to respect the inductive construction. We assume the usual truth conditions for logical connectives; although we will not discuss compound formulae in this paper.

$$\langle \mathcal{C}, s, i \rangle \models P \textit{ knows } \varphi$$

iff $\langle \mathcal{C}', s', i' \rangle \models \varphi$ at all $\langle \mathcal{C}', s', i' \rangle$ s.t. $\langle \mathcal{C}, s, i \rangle \sim_P \langle \mathcal{C}', s', i' \rangle$

This definition gives a strand semantics for knowledge in a distributed environment. However, we have not described what specific types of things φ might express. We can give semantics for formulae expressing the sending and receiving of messages without giving any more details about the model. But, before we do, we discuss the difference between the above knowledge semantics and some of those that have preceded it.

4.3.1 Discussion: What you see is what you get?

One of the especially tricky features of AT and SVO is how to represent the receipt of messages by a principal that were not understood, or worse, partially understood. The above semantics opts for simplicity in its respect of subtle epistemic intuitions.

To illustrate, if P receives $\{\{X\}_{K_2}\{Y\}_{K_3}\}_{K_1}$ and P has the keys K_1 and K_2 but not K_3, we may or may not not want to say that P knows that he has received $\{\{X\}_{K_2}\{Y\}_{K_3}\}_{K_1}$. Both AT and SVO adopt some notation to indicate those messages not recognized by P, essentially replacing $\{Y\}_{K_3}$ in this message with a placeholder for not-understood messages, i.e., those that cannot ultimately be tied back to plaintext. (SVO further differentiates specific not-understood messages so that, e.g., the same not-understood message can be recognized if seen again.)

This is summed up in SVO by the comprehension axiom that basically says that if P believes he sees $F(X)$, then he believes he sees X. 'F' here is meta-notation for any effectively one-one function such that either it or its inverse is computable in practice by P. This includes encryption and decryption with the relevant key treated as a parameter. The intuition behind this is that when P believes P received a message (as opposed to just receiving it) then P must understand what the message says, i.e., its structure.

The semantics we have described above does not respect this intuition. However, it respects another, somewhat contrary intuition, namely that P believes he received *this* message (whatever it is). The difference between

these two intuitions can be illustrated by means of the coin-flip example of section 1.1. If Alice sends to Bob $\{Heads\}_K$, then as long as Bob lacks K, he doesn't know that he has received $\{Heads\}_K$. In SVO and AT, this is represented by replacing $\{Heads\}_K$ with a placeholder.

On the understanding implied by the semantics above, a placeholder is not necessary. Bob knows he got $\{Heads\}_K$. He just doesn't know what that means. In particular he doesn't know therefore that he was sent *Heads* as opposed to *Tails*, or for that matter that this is an encrypted message as opposed to a random string. If on the other hand K were to be placed in Bob's key set at some point, at that point he would know that he got *Heads*, by the above truth conditions for *got* formulae. To some extent this intuition is captured in SVO by means of its distinct not-understood-message markers, but it still assumes that a principal understands all the structure in a message about which he has a belief. The above semantics may not capture the same epistemic subtleties as SVO, but it has a greater simplicity in this respect as well as a natural fit with the existing strand space constructs. We now return to the presentation of truth conditions.

Let M be an arbitrary message from our term algebra A.

$$\langle \mathcal{C}, s, i \rangle \models P \textit{ sent } M$$

iff there is a node $\langle t, j \rangle$ in $\mathcal{C}$ s.t. (i) $\mathrm{princ}(t) = P$, (ii) $\langle t, j \rangle \preceq \langle s, i \rangle$, and (iii) $\mathrm{term}(\langle t, j \rangle) = +M$

$$\langle \mathcal{C}, s, i \rangle \models P \textit{ received } M$$

iff there is a node $\langle t, j \rangle$ in $\mathcal{C}$ s.t. (i) $\mathrm{princ}(t) = P$, (ii) $\langle t, j \rangle \preceq \langle s, i \rangle$, and (iii) $\mathrm{term}(\langle t, j \rangle) = -M$

To give the truth conditions for other formulae, we must first spell out some of the structure of the term algebra and define a notion of submessage. The following definitions are taken from [22] and can also be found in the preceding strand space papers.

Assume the following:

- A set $\mathsf{T} \subseteq \mathsf{A}$ of texts (representing the atomic messages).
- A set $\mathsf{K} \subseteq \mathsf{A}$ of cryptographic keys disjoint from T, equipped with a unary operator $\mathsf{inv} : \mathsf{K} \to \mathsf{K}$.

  inv is injective; i.e., that it maps each member of a key pair for an asymmetric cryptosystem to the other; and that it maps a symmetric key to itself.

- Two binary operators

$$\mathsf{encr} : \mathsf{K} \times \mathsf{A} \to \mathsf{A}$$
$$\mathsf{join} : \mathsf{A} \times \mathsf{A} \to \mathsf{A}$$

We will follow notational conventions, some of which have already been mentioned, and write $\mathsf{inv}(K)$ as K^{-1}, $\mathsf{encr}(K, M)$ as $\{M\}_K$, and $\mathsf{join}(a, b)$ as $a\,b$. If k is a set of keys, k^{-1} denotes the set of inverses of elements of k.

The next assumption we make is that A is the algebra freely generated from T and K by the two operators encr and join. As noted in [22], this assumption has been commonly made in this area of research going back to [5]. As in [22] it is probably stronger than what we ultimately need but is pedagogically convenient. Amongst other things, it implies that encryptions and concatenations are unique and always distinct from each other and from T and K.

Central to the semantics of *said* formulae is the concept of an ideal. Interestingly, in the strand space papers, it was introduced to formulate general facts about the penetrator's capabilities; while in this paper, we will say virtually nothing about the nature of the penetrator.

Definition 4.3 If $\mathsf{k} \subseteq \mathsf{K}$, a k-ideal of A is a subset I of A such that for all $h \in I$, $g \in \mathsf{A}$ and $K \in \mathsf{k}$

(i) $h\,g, g\,h \in I$.

(ii) $\{h\}_K \in I$.

The smallest k-ideal containing h is denoted $I_{\mathsf{k}}[h]$.

The notion of ideal can be used to define a subterm relation $\sqsubset$ as follows [21].

Definition 4.4 Let $\mathsf{k} \subseteq \mathsf{K}$. $s \in \mathsf{A}$ is a k-*subterm of* $t \in \mathsf{A}$, $(s \sqsubset_{\mathsf{k}} t)$ iff $t \in I_{\mathsf{k}}[s]$.

If $\mathsf{k} = \mathsf{K}$ in this definition, then we say simply that s *is a subterm of* t, and write $s \sqsubset t$.

We now give truth conditions for *said* formulae

$$\langle \mathcal{C}, s, i \rangle \models P \textit{ said } M$$

iff there is a message M' s.t. $\langle \mathcal{C}, s, i \rangle \models P$ *sent* M' and $M \sqsubset_{\mathsf{k}} M'$ where k is the set of keys possessed by P at $\langle s, i \rangle$.

Notice that P is held accountable, e.g., for saying M at n, if he sends $\{M\}_K$ at $n' \preceq n$ and he has K at n, even if K was not in his key set until some n'' s.t. $n' \prec n'' \preceq n$.

A definition that does not occur in any of the strand space papers is that of a filter. In many contexts, filters are the duals of ideals. In our case, they are useful for giving semantics to *got* formulae, those that express the understood messages contained in received messages.

Definition 4.5 If $\mathsf{k} \subseteq \mathsf{K}$, a k-filter of A is a subset F of A such that for all $h, g \in \mathsf{A}$ and $K \in \mathsf{k}$

(i) $h\,g \in F$ implies $h \in F$ and $g \in F$

(ii) $\{h\}_K \in F$ implies $h \in F$ for $K^{-1} \in \mathsf{k}$

The smallest k-filter containing h is denoted $F_{\mathsf{k}}[h]$.

In general, the relation between filters and ideals is not so simple because, in public-key cryptography, one may have K and not have K^{-1}, or vice versa. However, in this paper we are limiting ourselves to the symmetric key case, $K = K^{-1}$. In this case there is a simple relation. (This relation also holds when both cognates of a public/private key pair are known.) It is easy to show that

Proposition 4.6 *For all sets of keys* k' *of the form* $\mathsf{k} \cup \mathsf{k}^{-1}$

$$g \in F_{\mathsf{k}'}[h] \quad \textit{iff} \quad h \in I_{\mathsf{k}'}[g].$$

Thus, for key sets k' of this form, by definition 4.4, $s \sqsubset_{\mathsf{k}'} t$ iff $s \in F_{\mathsf{k}'}[t]$. We can now give the truth conditions for *got* formulae. (We present them for the general case.)

$$\langle \mathcal{C}, s, i \rangle \models P \textit{ got } M$$

iff there is a message M' s.t. $\langle \mathcal{C}, s, i \rangle \models P$ *received* M' and $M \in F_{\mathsf{k}}[M']$ where k is the set of keys possessed by P at $\langle s, i \rangle$.

We can use the truth conditions for *said* and *got* formulae to further give the truth conditions for key goodness.

$$\langle \mathcal{C}, s, i \rangle \models P \overset{K}{\leftrightarrow} Q$$

iff, for all $\langle s', i' \rangle \in \mathcal{N}_{\mathcal{C}}$, $\langle \mathcal{C}, s', i' \rangle \models R$ *said* $\{M \textit{ from } Q\}_K$ implies either $\langle \mathcal{C}, s', i' \rangle \models R$ *received* $\{M \textit{ from } Q\}_K$, or $R = Q$ and $\langle \mathcal{C}, s', i' \rangle \models R$ *said* M. If $\langle \mathcal{C}, s', i' \rangle \models R$ *said* $\{M\}_K$ (instead of the stronger $\langle \mathcal{C}, s', i' \rangle \models R$ *said* $\{M \textit{ from } Q\}_K$), then $R \in \{P, Q\}$ (instead of the stronger $R = P$).

Note that these are the truth conditions from [17] with $\langle \mathcal{C}, s, i \rangle$ replacing $\langle r, t \rangle$ and $\langle \mathcal{C}, s', i' \rangle$ replacing $\langle r, t' \rangle$ throughout. This was itself based on the truth conditions for goodness given in [1].

Once we have a mechanism to express the beginning of the current epoch, we will be able to similarly dispatch the freshness and jurisdiction formulae. In order to do that, we must again confront the absence of a global concept of time. In the system models for possible world semantics of BAN-like logics, it was trivial to stipulate a global time t_0 and then define something as fresh if it was not said (by anyone) prior to t_0. We instead define a concept now as follows.

Definition 4.7 For any bundle $\mathcal{C}$, $\mathsf{now}_\mathcal{C} \subseteq \mathcal{N}_\mathcal{C}$, is a nonempty set of incomparable nodes (i.e., a nonempty set of nodes s.t. $n, n' \in \mathsf{now}_\mathcal{C}$ implies $n \not\preceq n'$ and $n' \not\preceq n$). For $n \in \mathcal{N}_\mathcal{C}$, we may write '$\mathsf{now}_\mathcal{C} \preceq n$' just in case there exists $n' \in \mathsf{now}_\mathcal{C}$ s.t. $n' \preceq n$. When it is clear from context which bundle is relevant, we will write simply 'now'.

Thus,

$$\langle \mathcal{C}, s, i \rangle \models \mathit{fresh}(M)$$

iff for all principals P, $\langle \mathcal{C}, s', i' \rangle \models P \ \mathit{said}\ M$ implies $\mathsf{now} \preceq \langle s', i' \rangle$.

The truth conditions for jurisdiction assume truth conditions for *says* formulae, which the definition of $\mathsf{now}_\mathcal{C}$ allows us to formulate.

$$\langle \mathcal{C}, s, i \rangle \models P \ \mathit{says}\ M$$

iff there is a message M' and a node $\langle t, j \rangle$ in $\mathcal{C}$ s.t. (i) $\mathrm{princ}(t) = P$, (ii) $\mathsf{now} \preceq \langle t, j \rangle \preceq \langle s, i \rangle$, (iii) $\mathrm{term}(\langle t, j \rangle) = +M'$, and (iv) $M \sqsubset_\mathsf{k} M'$ where k is the key set possessed by P at $\langle s, i \rangle$.

If φ is a formula.

$$\langle \mathcal{C}, s, i \rangle \models P \ \mathit{controls}\ \varphi$$

iff $\langle \mathcal{C}, s, i \rangle \models P \ \mathit{says}\ \varphi$ implies $\langle \mathcal{C}, s', i' \rangle \models \varphi$ for any $\langle s', i' \rangle$ s.t. $\mathsf{now} \preceq \langle s', i' \rangle$.

These conditions are similar to those in [1] and [17], *mutatis mutandis*. Notice that goodness is a condition that is constant across all points in the same bundle. And, jurisdiction and freshness are constant across all points in the present epoch. Notice also that jurisdiction is restricted to those messages that are formulae, rather than messages in general. This completes our presentation of truth conditions.

5 Conclusion

In this paper, we have set out a strand semantics for a BAN-style language. In the future we intend to set out an axiomatization that is sound with respect to this semantics. We also intend to connect this work with other approaches. The strand papers have already noted a connection to Paulson's inductive approach [13]. And, Meadows has observed connections between Paulson's inductive approach, ideals in strand spaces, and the construction of languages as used to prune an infinite search space down to manageable size in her NRL Protocol Analyzer[2]. In [9], an attempt was made to compare the computation model of AT with that of the NRL Protocol Analyzer (NPA). A number of open problems were described that needed to be resolved if they were to be ultimately integrated. In [18], a somewhat more optimistic comparison was made between the models underlying NPA and SVO. Given all of the above, it

[2] Personal communication of work in progress.

seems likely that strand spaces may provide the ultimate tie that binds BAN-style approaches to NPA. In addition to providing theoretical insight into the area, it is to be hoped that this will enable combining of the complementary applied advantages of each.

References

[1] Martín Abadi and Mark Tuttle. A Semantics for a Logic of Authentication. In *Proceedings of the Tenth ACM Symposium on Principles of Distributed Computing*, pages 201–216. ACM Press, August 1991.

[2] Michael Burrows, Martín Abadi, and Roger Needham. A Logic of Authentication. Research Report 39, Digital Systems Research Center, February 1989. Parts and versions of this material have been presented in various places including *ACM Transactions on Computer Systems*, 8(1): 18–36, Feb. 1990. All references herein are to the SRC Research Report 39 as revised Feb. 22, 1990.

[3] Pierre Bieber. A Logic of Communication in Hostile Environment. In *Proc. Computer Security Foundations Workshop III*, pages 14–22. IEEE Computer Society Press, Los Alamitos, California, June 1990.

[4] Richard DeMillo, Nancy Lynch, and Michael Merritt. Cryptographic Protocols. In *Proceedings of the Fourteenth ACM Symposium on the Theory of Computing*, pages 383–400. ACM Press, New York, 1982.

[5] D. Dolev and A. Yao. On the Security of Public Key Protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, March 1983.

[6] R. Fagin, J. Halpern, Y. Moses, and M. Vardi, *Reasoning About Knowledge*, The MIT Press, 1995.

[7] Li Gong, Roger Needham, and Raphael Yahalom. Reasoning about Belief in Cryptographic Protocols. In *Proceedings of the 1990 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 234–248. IEEE Computer Society Press, Los Alamitos, California, 1990.

[8] Jaakko Hintikka. *Knowledge and Belief: An Introduction to the Logic of Two Notions.* Cornell University Press, Ithaca, N.Y., 1962.

[9] C. Meadows. A Model of Computation for the NRL Protocol Analyzer. In *Proceedings of Computer Security Foundations Workshop VII*, pages 84–89. IEEE Computer Society Press, Los Alamitos, California, 1994.

[10] M. J. Merritt. Cryptographic Protocols. Ph.D. thesis, Georgia Institute of Technology, 1983.

[11] Michael Merritt and Pierre Wolper. States of Knowledge in Cryptographic Protocols, 1985. Unpublished Manuscript.

[12] Paul C. van Oorschot. Extending Cryptographic Logics of Belief to Key Agreement Protocols (Extended Abstract). In *Proceedings of the First ACM Conference on Computer and Communications Security*, pages 232–243, November 1993.

[13] Lawrence C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 1998. Also Report 443, Cambridge University Computer Lab.

[14] P. F. Syverson. Knowledge, Belief, and Semantics in the Analysis of Cryptographic Protocols. *Journal of Computer Security*, 1(3):317–334, 1992.

[15] Paul F. Syverson. Adding Time to a Logic of Authentication. In *Proceedings of the First ACM Conference on Computer and Communications Security*, pages 97–101. ACM Press, New York, November 1993.

[16] Paul F. Syverson and Paul C. van Oorschot. On Unifying Some Cryptographic Protocol Logics. In *Proceedings of the 1994 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 14–28. IEEE Computer Society Press, Los Alamitos, California, 1994.

[17] P. F. Syverson and P. C. van Oorschot. "A Unified Cryptographic Protocol Logic". Manuscript. (Preliminary versions of parts of this paper appeared in [12] and [16].)

[18] P. F. Syverson. Relating Two Models of Computation for Security Protocols. *Workshop on Formal Methods and Security Protocols* (affiliated with *LICS'98*), June 1998.

[19] F. J. Thayer Fábrega, J. C. Herzog, and J. D. Guttman. *Strand Spaces*. Technical Report, The MITRE Corporation, November 1997.

[20] F. J. Thayer Fábrega, J. C. Herzog, and J. D. Guttman. Strand Spaces: Why is a Security Protocol Correct?. In *Proceedings of the 1998 IEEE Symposium on Security and Privacy*, pages 160–171. IEEE Computer Society Press, 1998.

[21] F. J. Thayer Fábrega, J. C. Herzog, and J. D. Guttman. Honest Ideals on Strand Spaces. In *Proceedings of the 1998 IEEE Computer Security Foundations Workshop*, pages 66–77. IEEE Computer Society Press, 1998.

[22] F. J. Thayer Fábrega, J. C. Herzog, and J. D. Guttman. Strand Spaces: Proving Security Protocols Correct. Forthcoming in *Journal of Computer Security*.

[23] M.-J. Toussaint. Separating the Specification and Implementation Phases in Cryptology. In *Computer Security - ESORICS 92*, volume LMCS 638, pages 77–102. Springer-Verlag, 1992.